

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) A queue comprising:
 - a first queue configured to enqueue and dequeue data units, the first queue including a plurality of parallel sub-queues that queue a plurality of parallel data units;
 - a second queue configured to receive data units from the first queue when the first queue has data units available to be dequeued, the second queue including a first buffer configured to store a first set of the parallel data units and a second buffer configured to store a second set of the parallel data units, the second queue outputting up to a predetermined number of the first set of the parallel data units in a clock cycle and, when a number of data units in the first set of parallel data units is less than the predetermined number, output one or more of the second set of parallel data units in parallel with the first set of the parallel data units; and
 - bypass logic coupled to the second queue, the bypass logic configured to bypass the first queue and to forward data units to the second queue when the second queue is ready to receive data units and the first queue is empty.
2. (previously presented) The queue of claim 1, wherein the data units consists of memory access requests.

3. (canceled)

4. (canceled)

5. (previously presented) The queue of claim 1, further comprising:
an encoding component coupled to the bypass logic and the first and
second buffers, the encoding component configured to read data units from the
first and second buffers, wherein the encoding component gives data units in the
first buffer higher priority than data units in the second buffer.

6. (canceled)

7. (currently amended) A queue comprising: The queue of claim 1, further comprising:

a first queue configured to enqueue and dequeue data units, the first queue including a plurality of parallel sub-queues that queue a plurality of parallel data units;

a second queue configured to receive data units from the first queue when the first queue has data units available to be dequeued, the second queue including a first buffer configured to store a first set of the parallel data units and a second buffer configured to store a second set of the parallel data units;

bypass logic coupled to the second queue, the bypass logic configured to bypass the first queue and to forward data units to the second queue when the second queue is ready to receive data units and the first queue is empty; and

masking logic coupled to the output of the first and second buffers, the masking logic configured to restore data units to the first and second buffer that were not output from the first and second buffers.

8. (previously presented) A method of masking latency in a device, the method comprising:

receiving incoming data items for a queue that include a plurality of data items that are input to the queue for each cycle of the queue;

forwarding the incoming data items to a buffer when the queue is empty and the buffer is free to receive data items, wherein the buffer includes a first buffer and a second buffer, and wherein higher priority data items are stored in the first buffer and lower priority data items are stored in the second buffer;

enqueueing the incoming data items in the queue when the queue contains data items or the buffer is not free to receive data items;

dequeuing data items from the queue to the buffer when the buffer is free to receive data items; and

transmitting the data items from the buffer as the output of the device.

9. (original) The method of claim 8, wherein the data items are memory access requests.

10. (canceled)
11. (canceled)
12. (previously presented) A method of masking latency in a queuing device, the method comprising:
 - receiving incoming data items for a queue;
 - forwarding the incoming data items to a buffer when the queue is empty and the buffer is free to receive data items, the buffer including a first buffer and a second buffer, and wherein higher priority data items are stored in the first buffer and lower priority data items are stored in the second buffer;
 - enqueueing the incoming data items in the queue when the queue contains data items or the buffer is not free to receive data items;
 - dequeuing data items from the queue to the buffer when the buffer is free to receive data items; and
 - transmitting the data items from the buffer as the output of the queuing device,
 - wherein the data items in the second buffer are moved to the first buffer when the first buffer is free to receive data items.
13. (previously presented) The method of claim 8, wherein two data items are transferred from the first and second buffer per cycle as the output of

the queuing device whenever the first and second buffer contain at least two data items.

Claims 14-19 (canceled)

20. (previously presented) A device comprising:
 - means for receiving incoming data that includes a plurality of data for each cycle;
 - means for buffering the data before transmitting the data in a first buffer and a second buffer, in which higher priority data is stored in the first buffer and lower priority data is stored in the second buffer;
 - queue means;
 - means for forwarding the received incoming data to the means for buffering when the queue means is empty and the means for buffering is free to receive data;
 - means for enqueueing a plurality of the incoming data to the queue means, in a cycle of the queue means, when the queue means contains data or the means for buffering is not free to receive data; and
 - means for dequeuing data from the queue means to the means for buffering when the means for buffering is free to receive data.
21. (previously presented) The device of claim 20, wherein the data consist of memory requests.

Claims 22-26 (canceled)

27. (previously presented) The method of claim 8, wherein transmitting the data items from the buffer includes:
transmitting up to a predetermined number of data items from the first buffer in a first cycle of the queue and, when the first buffer does not include the predetermined number of data items, transmitting additional data items from the second buffer, up to the predetermined number of data items, in the first cycle of the queue.